SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AIM 1189

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Machine Recognition as Representation and Search

5. TYPE OF REPORT & PERIOD COVERED
Memorandum

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Feng Zhao

8. CONTRACT OR GRANT NUMBER(s)
N00014-89-J-3202

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, MA 02139

10. PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12. REPORT DATE
December 1989

13. NUMBER OF PAGES
40

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, VA 22217

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES
None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer vision
Representation
Search control
Object modeling
Consistent labelfng
Model-based recognition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Generality, representation, and control have been the central issues in machine recognition. Model-based recognition is the search for consistent matches of the model and image features. We present a comparative framework for the evaluation of different approaches, particularly those of ACRONYM, RAF, and Ikeuchi et al. The strengths and weaknesses of these

(continued on back)

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued:

approaches are discussed and compared and the remedies are suggested. Various tradeoffs made in the implementations are analyzed with respect to the systems' intended task-domains. The requirements for a versatile recognition system are motivated. Several directions for future research are pointed out.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1189 December 1989

Machine Recognition as Representation and Search

Feng Zhao

Abstract

Generality, representation, and control have been the central issues in machine recognition. Model-based recognition is the search for consistent matches of the model and image features. We present a comparative framework for the evaluation of different approaches, particularly those of ACRONYM, RAF, and Ikeuchi et al. The strengths and weaknesses of these approaches are discussed and compared and the remedies are suggested. Various tradeoffs made in the implementations are analyzed with respect to the systems' intended task-domains. The requirements for a versatile recognition system are motivated. Several directions for future research are pointed out.

Keywords: computer vision, model-based recognition, representation, object modeling, search control, consistent labeling.

This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-89-J-3202.

Machine Recognition as Representation and Search

Feng Zhao

December 1989

Abstract

Generality, representation, and control have been the central issues in machine recognition. Model-based recognition is the search for consistent matches of the model and image features. We present a comparative framework for the evaluation of different approaches, particularly those of ACRONYM, RAF, and Ikeuchi et al. The strengths and weaknesses of these approaches are discussed and compared and the remedies are suggested. Various tradeoffs made in the implementations are analyzed with respect to the systems' intended task-domains. The requirements for a versatile recognition system are motivated. Several directions for future research are pointed out.

Keywords: computer vision, model-based recognition, representation, object modeling, search control, consistent labeling.

1 Introduction

The task of making a computer "see" things has fascinated many researchers for several decades. The task is challenging enough for its own sake. The results can shed light on how human vision works, and can be used in many practical applications. Although research in neurophysiology has revealed much biological evidence that might suggest how the human vision system works, there has been little progress towards general understanding of human vision. David Marr [72], on the other hand, approached the problem from a computational point of view and defined vision as achieving a particular visual task. The task can be analyzed and implemented computationally. This task-oriented, computational approach has stimulated much creative research, from which many recent approaches have stemmed.

1.1 Machine recognition

A recognition system is usually part of a larger system that senses the environment, interprets the sensed data, understands the objects in the scene, and acts upon the world. The goal of machine recognition is thus, given sensed data and *a priori* knowledge about the world, to "see" what objects are in the scene and, if there are any, the precise positions and orientations of the objects. Besl and Jain [3] give a mathematical formulation of machine recognition, defined as an inverse mapping of scene to image projection.

This paper reviews three different approaches [16, 33, 60] to object recognition in computer vision. These approaches, along with other work, constitute an important class of machine recognition methods. They are instances of the successful paradigm in machine recognition — model-based recognition [3, 6, 19] — but differ in their representation of objects, handling of controls, and intended task domains. In this paradigm an observed object is recognized as being an instance of a model if their respective primitives are pairwise consistently matched [15]. The paradigm has its origin in the work of scene analysis [77, 41, 49, 20, 85]. The representation of objects and images, the selection of data, and the control of the matching process have been the central issues in model-based machine recognition and have been actively researched.

The ACRONYM system [16] is a general, domain independent system. It recognizes objects in the image and determines the positions of the objects. Geometric constraints are used to relate parts of objects. The RAF system [33] handles overlapping objects in a polyhedral world. It searches an interpretation tree that represents all the matches to be considered. Local geometric constraints are used to drastically reduce the size of search. Ikeuchi et al.'s system [60] works in bin-picking task domains. It assumes that only instances of one object are present in the image and does not handle occlusion. On the other hand, Ikeuchi et al. present a complete methodology of automatic programming for machine recognition.

In this paper we will motivate the central issues of machine recognition — GENERALITY, REPRESENTATION, and CONTROL — and cast recognition as representation and control of search, driven by its intended generality. We will then discuss how different approaches can be compared in a framework defined by the three axes: generality, representation, and control. We will analyze various tradeoffs made in the implementations with respect to systems' task domains.

1.2 Putting machine recognition into a larger context

There has been extensive research in the classical areas of image processing, early vision, pattern recognition, and scene analysis.

Image processing takes images and produces "enhanced" images that are useful as inputs to vision systems. Early vision systems produce descriptions of the input image, for example intensity changes like edges [45] and needle maps [70, 47], and group them into regions based onto some similarity measures [72]. Recognition systems use the outputs of early vision systems to identify scene objects and determine their positions and orientations [3]. Many techniques of early vision systems have been used in machine recognition.

Machine recognition is closest to pattern recognition and scene analysis. Pattern recognition [23] classifies an object from a vector of primitive features that are measurements of some global characteristics of the object. These measurements include area, perimeter, center of mass, moments of inertia (for orientation), ratio of maximum and minimum inertia (for elongation). The feature vectors form a feature space. The hope is that the feature vectors of different classes of objects are spatially apart in the feature space so that some simple techniques like nearest neighbor method can be used to cluster features into distinct classes. Classification of an unknown object is to determine which cluster it belongs to. The closeness of a feature point to a class is measured by some statistical quantities. Therefore pattern recognition proceeds in two steps, feature extraction and pattern classification. The features used are generally global in nature; Many rich local geometric relations are left out. Also, it is hard to find features that are invariant with respect to variations, such as those in viewing directions, and extract them reliably. Many techniques developed for feature extraction and classification have been used in machine recognition.

Scene analysis [23, 87, 86] interprets a description of some scene and assigns scene objects with known labels. It consistently labels objects in the scene, subject to constraints derived from the objects and images, to give a meaningful interpretation of the scene. The research on line labeling of polyhedral scenes started with Robert's seminal work [77], and was followed by Guzman [41], Huffman [49], Clowes [20], and Waltz [85]. We will discuss in some details the work on line drawing interpretation in Section 3.4.1. Although it does not concern the problem of producing reliable picture descriptions, the line drawings, the techniques of feature representations, and the control of consistent labeling have been extensively used in many approaches to machine recognition today.

2 A Comparative Framework for Machine Recognition

2.1 Model-based recognition

Model-based recognition assumes the existence of a library of models. Models can be specified in terms of geometric properties of the objects, or with other properties if available. Models and images can be described by a set of features and constraints that relate the features. The features and constraints form a network. Recognition is the search for consistent matches (which constitute the identification of the image object) subject to constraints, in the space of all possible matches between model features and image features, and the determination of a transformation from the model to the image object.

Three basic questions have often been asked in machine recognition:

1. What constitute useful features and constraints?
2. How can features and constraints be reliably extracted?
3. How can features and constraints be used for recognition?

In this paper, we will focus on high-level aspects of these questions. We concentrate mainly on the questions one and three. Although much of the success of machine recognition depends on the availability of features and constraints provided by early vision modules, in this paper we emphasize the form and the use of features and constraints.

We identify the three central issues of machine recognition: GENERALITY, REPRESENTATION, and CONTROL, with respect to which a comparative framework is defined. We compare different approaches to machine recognition in this comparative framework, see how each of their intended generalities defines a subspace, and discuss how tradeoffs are made within the subspace and across several subspaces. Figure 1 shows this comparative framework. Each subspace is identified with a plane.

We will evaluate a recognition system with its two components, the representation and the control of search, and discuss how the design of the system is driven by its intended generality. The following metrics are used for the evaluation of each approach:

- performance on its intended task,
- sensitivity to noise and occlusion, and
- its task-domain-independent mechanisms that can be generalized.

If we view the task of recognition as being composed of two parts: the identification of an image object and the localization (position and orientation) of the object, the three approaches that we will discuss in details use more or less the same method: hypothesize a match between the image object and a model and verify that the match actually produces a legal interpretation of the image. Other methods like the Alignment Method [83] proceed by hypothesizing the position and orientation first, and use the hypothesized model to constrain the search in the object identification. The theme of all these approaches is the same: hypothesize and verify. They differ in how much and how strong the hypotheses are.

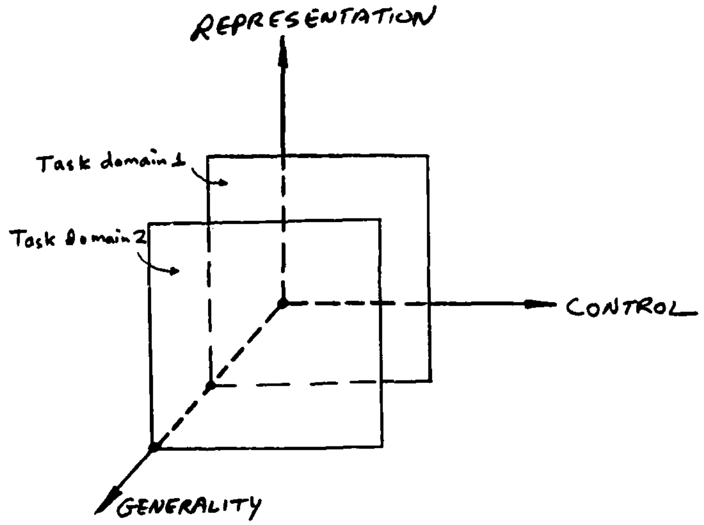

Figure 1: A comparative framework defined by generality, representation, and control.

2.2 The comparative framework: generality, representation, and control

2.2.1 Generality

There have been two contrasting trends in approaches to machine recognition:

- understanding the principles of human vision and general methodologies of vision systems;
- achieving high performance for industrial applications.

The first school of approaches attempt to answer the question of how the human vision system works. They hope to formulate a general theory and methodology for solving machine recognition problems. The systems thus built embody mostly domain independent principles. The other group tries to find solutions to many specialized industrial applications. It deals with restricted situations that demand high accuracy and speed, such as robot arms picking an object from a pile on some flat surface.

1. *The Dilemma: Generality or Performance?*

   It is not uncommon to see that systems aiming for generality either suffer from complexity and system bugs, or work only on toy problems, and often do not deliver their promises. On the other hand many recognition systems working well on specific industrial applications are overly restrictive and do not generalize to a slightly different domain. The one system/one application situation requires experts to modify the system significantly in order to accommodate changes in task domains. Some people pursue the middle road that uses some psychological and neurophysiological evidence to guide the design of a vision system [72]. The hope is to build a system that has some generality and at the same time works well on a class of real problems.

2. *The Task Domains:*

   A task domain imposes strong restriction on the images and object models, and dictates what the system should achieve. The task also determines what tradeoffs would be made, such as efficiency vs. generality and efficiency vs. accuracy. The use of search cutoff in RAF (i.e., thresholding) and precompiling constraints into a table in Ikeuchi et al. are all instances of these tradeoffs.

   The input to a recognition system consists of one or more images, each of which is either a 2D image, a 2-$\frac{1}{2}$D map, or some other preprocessed image containing useful feature descriptions. The output is what can be used to perform a task, for example, picking the identified object from a bin. Clearly the output is task dependent. The divergence of criteria for the outputs of recognition systems is part of the reason that comparing different approaches is very difficult.

3. *The Answer:*

   A system is usually designed for solving a special class of problems. Ideally a good system, although developed for specific problems, should embody principles sufficiently independent of the particular applications so that they are applicable to other domains as well, without too severe degradation in performance. The system can be part of a more general system that handles more complex problems. The answer to this dilemma is to have some separation between domain-dependent modules and domain-independent modules of the system. The underlying theory of the system and the tradeoffs in its working implementation need to be separated as well. Some tradeoffs result from limitations of technology. Others are from the lack of either the understanding of or the availability of high quality low level modules. Ad hoc heuristics are introduced in order to make the theory work. However systems bearing good separation can be easily improved later.

2.2.2 Representation

Marr [70] characterized the criteria and design of representations of geometric shapes. He used three criteria for judging the usefulness of a representation:

1. accessibility — whether a description can be computed easily;
2. scope and uniqueness — the class of shapes it represents and the uniqueness of the description for each shape;

3. stability and sensitivity — the resolution with respect to small variations in shape parameters.

Marr argued that a good representation should be coordinate system free (that is, use an object-centered, as opposed to viewer-centered, coordinate system), consist of primitives with various sizes, and admit a modular organization. He also discussed the use of indexing techniques to show how to hierarchically organize the object parts and the use of additional visual cues to overcome the difficulty of recovering the shape due to partial occlusions.

In the domain of recognition, representations are necessary for images, models, and sensors[1]. The central issue is how to choose and represent features and constraints, since the choice and the description of features ınd constraints affect the performance (accuracy, efficiency, etc.) of recognition algorithms. Most past research has concentrated on representing models and images, while little attention has been paid to modeling sensors, with the noble exception of Ikeuchi et al. [62]. Without an analysis of sensors it is hard to discuss the features quantitatively.

To summarize, the following are desirable properties of a representation:

- The features and the associated constraints capture the characteristics of object models, images, and sensors. Restrictions and assumptions on the image, model and sensor domains are explicitly reflected. There exists some metric of feature quality which can be used to decide which features to choose and in what order they are used, so that the search for consistent matches is most efficiently carried out. The representation also facilitates the acquisition of models.
- The representations of images and models admit tractable and reliable computation of their features. The sensor representation lends itself to easy analysis of errors. There exist robust algorithms for extracting features and precise models for describing constraints.
- When there are a variety of features and constraints, the representation has a modular organization that admits some form of indexing into the model library.
- There exist efficient mechanisms to use the features and constraints for the matching process of recognition tasks. We will discuss this in details in the following section on control.

2.2.3 Control

In machine recognition, *control* refers to the use of features and constraints of the representation in the recognition process. A brute-force search for matches between the observed object and a model will develop a huge search tree, since there is an enormous semantic

[1] For example, camera models and light models.

distance between the input and the output of a recognition system. The resulted matches are also sensitive to small variations in object and image parameters.

One important control paradigm is consistent labeling[2], which has its roots in the work of interpreting picture line drawings [41, 49, 20, 85]. It originated from Huffman and Clowes's work on line drawing labeling [49, 20]. This paradigm works as follows: given a catalogue of image labels and consistency rules, the algorithm develops a tree by assigning each scene feature with a label and backtracking when the assignment is inconsistent with what is already known. Waltz [85] uses a filtering process to eliminate locally inconsistent label assignments by the consistency rules and then conducts a global tree search. As the result of the filtering, the size of the tree search is reduced.

Successful recognition can be viewed as a consistent labeling of the image objects subject to some (often geometric) constraints. One common control strategy is hypothesize-and-verify, that is, hypothesize a match of a model to the observed object and verify if the match gives a legal interpretation to the image. Various techniques are used to further reduce the need for search. Attention-focus [7] and preliminary grouping [63, 64] are two such strategies. How global knowledge of models is used also affects the performance.

Approaches based on consistent labeling have several problems that deserve special attention. A complete catalogue of legal labels might contain too many entries (e.g. several million) that are too expensive to search through. The problem demands the use of some form of indexing into the catalogue in order to achieve any reasonable performance. Hierarchical grouping of entities can reduce the search time. The question is how to organize the catalogue that reflects the semantic structure of the world. Worse, it is possible that a complete catalogue of legal labels is either technically difficult to obtain or physically non-existent. For picture line drawings where a junction is formed by arbitrary number of edges, Huffman [51] showed that a decision procedure can instead be used to eliminate impossible labelings. However, in general whether we can find some test for arbitrary objects is still a question.

3 Three Model-based Recognition Systems

This section reviews the three approaches of ACRONYM, RAF, and Ikeuchi et al. to machine recognition. We use the requirements for representation and control and the principles of generality to evaluate each approach. The task domain and the intended level of performance of each of three systems is different from each other, ranging from general to specific. The implementation of each system is greatly influenced by its task to accomplish. Many task-domain specific tradeoffs are made along the way from the general theory of each system to its working implementation.

[2] An alternative to consistent labeling is the relaxation method. Simply stated, the relaxation method is a spring-loaded template matching. Interested readers should refer to [21] for more details on the relaxation method.

There are commonalities among them. Although each system chooses somewhat different representation and description of features and a different set of constraints, and uses somewhat different techniques to reduce the size of search space, all three systems use more or less the same control strategy, i.e., consistent labeling. These approaches search for possible interpretations subject to their own set of constraints. The two principal components of the three systems are the representation of features and constraints and the search under constraints.

3.1 ACRONYM

3.1.1 Overview

ACRONYM was developed by Brooks et al. [16] at Stanford in the late 70's and early 80's. It was intended to be a general, domain-independent, model-based recognition system. More precisely, it is a working implementation of a general recognition theory of prediction from models and interpretation of images. It uses a volumetric representation of objects and a domain-independent geometric and algebraic constraint manipulation system. The system has been tested on several aerial images of airfields and has successfully recognized certain wide-bodied jets.

ACRONYM is equipped with a library of object models. The geometric object models are specified in terms of a few simple primitives and their relations. Given a 2D image, ACRONYM identifies instances of object model classes, determines their location and orientation in world coordinates, makes subclass identification if possible, and determines location and orientation of camera if necessary.

Brooks [15] views recognition as a task of finding consistent matches between elements of an image and those of a model. The question is at what level the matches take place. He argues that, in contrast to the traditional image-driven view of vision as inverse optics [72], recognition can proceed from the model in coarse to fine levels. Figure 2 shows the levels of representations of the model and the image and the correspondence between them at each level. The appropriate descriptions of the model are predicted from the model and are matched against the descriptions of the image at the same level.

The approach of ACRONYM to image understanding relies on four components: object models, prediction from models, interpretation of image descriptions in terms of model prediction, and descriptions of images. These four modules are associated with four data structures: object graph and restriction graph, prediction graph, interpretation graph, and picture graph. Figure 3 shows the four modules, data structures, and data flow paths of ACRONYM. ACRONYM deals with only the first three components. It does not use the rich image descriptions available. Instead it uses a simple and primitive system to produce descriptions of images. The three components are tied together in ACRONYM by two threads: geometric models and relationships, and algebraic constraints.

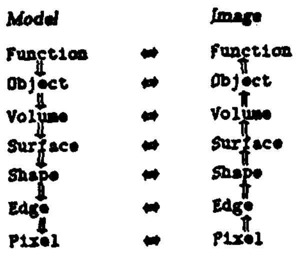

Figure 2: Levels of representation (from Brooks)

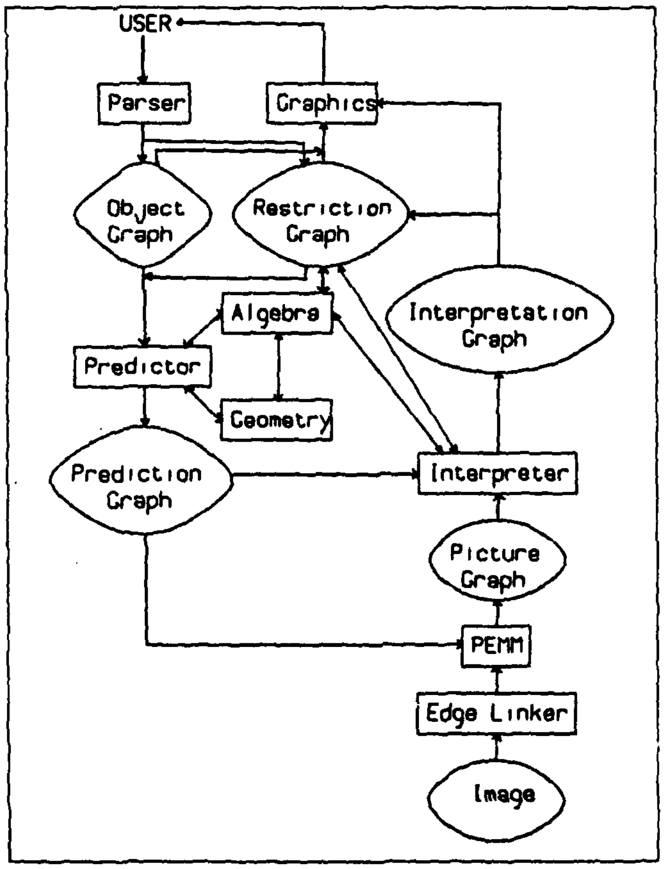

Figure 3: The ACRONYM System (from Brooks)

3.1.2 Evaluation

In the following discussions, the ACRONYM system is evaluated in terms of its four modules and the interactions among them.

1. *Object models*

   ACRONYM uses view-independent volumetric representation for the objects. The volume primitives are generalized cones [5]. The geometric representation models the objects and their spatial relations. Generic object classes and specific objects are modeled by class and subclass relations through algebraic specifications. Variations of size, structure and spatial relations within object classes are modeled. Volumetric models and spatial relations are represented by the Object Graph (Figure 3). The nodes are volume elements and arcs are spatial relations and subpart relations. A user specifies models through a text-based description language. A graphics module provides the user with feedback during the process of modeling.

   A generalized cone is produced by sweeping a cross section along a spine through space. The cross section is kept at a constant angle while sweeping along the spine and is deformed as specified by a sweeping rule. For example, an airplane is modeled by a fuselage with two wings attached to it, each of which is a generalized cone. Many man-made objects can be decomposed into volume primitives in a few ways.

   However, generalized cones with arbitrary deformation functions and general shapes of cross section will be computationally intractable to deal with. Certain tradeoffs have to be made for the purpose of realistic implementation. In ACRONYM, the generalized cones are therefore further restricted. The cross section is bounded by straight lines and circular arcs, spines are piece-wise linear or circular, and the sweeping rule is linear and continuous. As a result, the relatively simple geometric representation facilitates the prediction of features.

   One of the requirements for representation is that the representation is unique, that is, it admits a unique decomposition of an object into parts. If there is more than one decomposition for an object, then the image and the model may be decomposed incompatibly and fail to match correctly. Although many man-made objects admit decomposition into subparts, in general it is hard to characterize real world objects with a few such generic parts. Worse, many objects do not have a natural decomposition. For example, it is awkward to think about a bushy object in terms of generic subparts. Bushy objects are more effectively dealt with using other types of descriptions such as edges.

   ACRONYM models a complex object by a set of generalized cones. The spatial relations of the hierarchy of object parts are represented by subpart hierarchy and an affixment tree. The class/subclass relations capture the commonality of class members and variations across class members. This is advantageous for cases where only partial recognition can be obtained. For example, an unknown car can be recognized as a member of a compact car and leaves the specific model type undetermined. Additional information, such as the local shape of the trunk, the color, etc. will help to disambiguate among different models.

2. *Prediction*

   In ACRONYM features are predicted by geometric reasoning techniques. The geometric reasoning mechanism combined with the geometric and algebraic representations makes ACRONYM powerful and flexible in predicting features.

   Object features in ACRONYM are invariant with respect to both object class and viewpoint. They include shapes (ribbons and ellipses) and 2D spatial relations of shapes in the image, and are specified by various constraints. Constraints are managed by a constraint manipulation system (CMS). The object models are analyzed for variations, structure, and spatial relations in the object model classes and its expected features are predicted. The result is the Prediction Graph (Figure 3) whose nodes specify predictions of image features, and whose arcs specify relations of image features. The prediction graph tells the matcher how to find instances. It provides a coarse filter for hypothesizing matches of object and image features. The prediction graph also contains instructions on how to use the measurement of image features to deduce 3D information about original models. Note that the prediction graph is a symbolic summary of expected image appearances of models. It does not produce image appearances of instances of models.

   The flow of control in prediction is backward-chained. A breadth walk down the subpart hierarchy of object models predicts shapes at each level and partially interprets the image. Refined prediction is made at the next level. The domain independent constraint manipulation system (CMS) deduces and propagates constraints. As Brooks noted much of ACRONYM's expertise comes from the generality of the CMS to handle a variety of constraints [16], however at the expense of being slow. Although it has successfully interpreted some aerial images, applying ACRONYM to perform some realistic applications would require significant improvement in its speed. One way to achieve this is to prespecify the constraints, such as the angle between wings and fuselage of a certain type of jets, then compile the constraints into tables. Runtime constraints manipulation would merely be a table lookup. Grimson and Lozano-Perez [33] explored this idea in their recognition system RAF.

3. *Interpretation*

   The interpretation proceeds by merging local hypothesized matches subject to consistently derived constraints about variations of size, structure, and spatial relations. The candidate image features are provided by image descriptive process and are matched to predicted object features. If the match is consistent with what is already known about the model, it puts additional constraints on the parameters of the 3D model.

The descriptive process is invoked repeatedly. At first the multiple invocations search for different image features to obtain a coarse image interpretation. Later invocations search for small areas of image for particular features in order to get detailed object class identification and three dimensional quantification of objects. The prediction, description, and interpretation proceed concurrently from coarse object subpart and class interpretations of images to fine distinctions among object subclasses and more precise 3D information about objects, as finer and finer details of objects are identified.

The mechanism here is essentially consistent labeling. It hypothesizes a global match, predicts additional evidence from coarse to fine, matches against that of images, and discards inconsistent matches. The constraints on the parameters of the hypothesized global match are successively tightened along the way by additional supporting evidence. Although at the high level it has some flavor of the parameter relaxation method [21], it is closer to consistent labeling. With the hypothesized match, the search is constrained by the knowledge about the model. Again, this is made possible by the class/subclass hierarchy. In some sense ACRONYM is close to the alignment method [83] which hypothesizes a match by transforming the model to the observed object and uses the model to constrain the search, although the constraints of ACRONYM are much looser.

4. *Image description*

ACRONYM uses a low level feature description module to link together primitive image edges. The local image feature descriptions and matches with the models are guided by the predictions which provides goal direction to an edge-linking algorithm [11]. The edge linker is directed to preliminarily group together those edge segments that are likely to come from the same object in the image.

This initial grouping reduces the search in the interpretation process. Jacobs [64] has shown that similar techniques of preprocessing image features have the effect of reducing search time and increasing accuracy of matches. The use of edging linking process explains partially why ACRONYM does so well even though the initial image descriptions are very poor. The drawback of ACRONYM's edge linking preprocessing is that the edge linker uses some domain specific heuristics of the aerial images of airfields, which are not generalizable to other applications, to link the edge descriptions of poor quality.

In summary, ACRONYM was designed to investigate the use of models that are independent of particular descriptive processes and to develop general mechanisms for geometric reasoning and symbolic constraint propagation. Its strength comes mainly from (1) the simple geometric models and relations and the geometric reasoning system used to predict and guide interpretation; (2) the algebraic constraints and the powerful constraint manipulation system that model class relations, give quantitative aspect to predictions, prune incorrect interpretations, and provide 3D information about hypothesized objects;

and (3) the initial edge linking process that reduce the search and increase the accuracy of the interpretation.

The philosophy of ACRONYM capitalizes only on knowledge of geometric properties of objects and general mechanism of constraint manipulation. It assumes very little dependence on image descriptions. We have seen in the above analysis that in order for ACRONYM to achieve realistic performance on real images some tradeoffs are made in the implementation. This is somewhat a departure from the general philosophy its designers had in mind at the beginning. The approach of Grimson and Lozano-Perez discussed in the next section addresses issues of obtaining high performance on moderately complicated images, by trading off some generality of the system.

ACRONYM does not pay special attention to occlusions and shadows. To extend the system to handle these situations it should explore the rich knowledge of image descriptions. Other type of visual cues such as that of image formation may also be used, in addition to the geometric knowledge of objects, to enhance performance of the system. However, in general we still don't know how to integrate different visual cues into a single frame that uses them discretionarily and invokes them at appropriate times.

3.2 RAF

3.2.1 Overview

Grimson and Lozano-Perez [33] of MIT reported an approach to model-based recognition based on searching in the form of an interpretation tree subject to local geometric constraints. Their implementation of the method is the RAF[3] system.

The approach of Grimson and Lozano-Perez is clearly an instance of consistent labeling. The task is to identify and locate objects in cluttered environments, that is, to find what objects are in the scene and where they are, given a library of known object models. Recognition consists of two steps: generation of feasible interpretations (hypothesizing step) and model test (verification step). The first step is to develop an interpretation tree and search for correct matches in the interpretation tree. It identifies what objects are in the scene by relating the observed objects with the known models. Local geometric constraints are used to prune the tree search. The second step checks for global consistency. It also finds the positions and orientations of the identified objects in the image.

The method assumes that objects are approximated by a collection of polyhedral models. The adequacy of the model is discussed in a later paper [34]. Curved objects are hard to model in this way. The polyhedral model is also not stable. Small variations in parameters of a curved object can result in a significantly different polyhedral approximation and lead to erroneous matches. RAF works in the environment of robot sensing. The input

[3] RAF stands for Recognition and Attitude Finder.

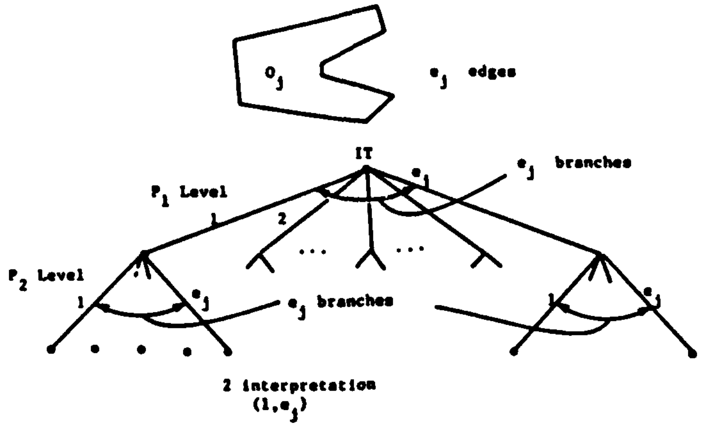

Figure 4: Interpretation Tree (from Grimson et al.)

sensed data to RAF consist of positions and the associated normals of object surface. The data can be sparse, which is typical of tactile sensing.

In an earlier version of RAF [30], the sensed data are assumed to come from a single object, either 2D or 3D, and no occlusion is allowed. The recognition is formulated as a search of an interpretation tree. The interpretation tree is constructed in a depth first fashion by assigning each sensed data point onto every face of a model. A tree branch is developed by pairing a data point with a face. Every path to leaves of the tree is a possible interpretation of the sensed data with respect to the model. Then each possible interpretation is tested against the model to see all the points are on and within the faces of the model. Let s be the number of data points and n be the number of faces of the object: the interpretation tree is s levels deep and each node has branching degree of n. Figure 4 shows an interpretation tree for a 2D polygonal object model. RAF generates an interpretation tree for each model.

A brute-force search in the interpretation stage is too expensive to be feasible, since the number of possible interpretations that the model is tested against is the number of tree leaves, or n^s. Three types of geometric constraints [30] are considered to reduce the size of search space: (1) the distance constraint for distances between faces; (2) the angle constraint for angles between face normals; and (3) the direction constraint for angles of vectors between sensed points. For example, the possible distance between a pair of points on object faces has to be within the legal range specified by the geometric model of the object and the error model of measurements.

RAF is later generalized to handle occluded objects [33]. To deal with the situation where input data may come from multiple objects, a null branch is created below each interpretation tree node. Assigning a data point to this node is equivalent to discarding the point as inconsistent with the model (note that one interpretation tree is constructed per model). In addition to the three powerful geometric constraints, further mechanisms are used to limit the search:

- preprocessing to find extended features such as edges;

- heuristically guided search and premature search termination;
- Hough clustering[4] to preselect likely subspaces.

3.2.2 Evaluation

1. *Mechanisms to improve the performance:*

   Simulations of RAF on several 2D and 3D examples have shown the power of local geometric constraints in reducing the effort of the search. RAF has since been extended to handle scaled objects [35], curved objects [34], as well as objects that contain movable subparts [35]. Preliminary results show that the search increases significantly with the complexity of the scene. It should be noted that each extension is developed under the assumption that only one variation, for example scaled, curved, or movable object, occurs at a time. Future extension will be to the cases where more than one variation in object is allowed. It will be interesting to see how the techniques developed for individual extensions can be combined and how the performance degrades. Another direction is to explore additional constraints that capture characteristics of objects in these situations.

   The performance of RAF is greatly improved by the use of extended features such as edges for 3D samples. Point-like data are shown to be inefficient when objects are occluded. The initial segmentation and grouping to obtain extended features drastically reduce the size of search space. Another technique is the use of Hough clustering to prefilter candidates for hypotheses so that the attention of the system can focus on "good" ones. This method helps reduce the search time, however at the expense of introducing additional matching errors.

   The introduction of "null branches" increases exponentially the number of feasible interpretations generated and therefore increases greatly the complexity of the model test. Heuristic search ordering is used to guided the search towards "good" interpretations by a measure of quality. The search is cut off as soon as the measure reaches an acceptable level. The heuristic technique — the premature termination of the search — is essential in reducing the search time.

   In the experiments, a simple heuristic of ordering picks more distant points first and puts them at the early stage of the paring process. This puts the most effective constraints at the beginning and results in pruning out entire subtrees at as early a stage in the tree generation as possible. The interesting question is, when different type of constraints become available, in what order the constraints should be applied to most effectively prune the tree search. The key is to have a metric of weighing constraints according to their power in constraining the search.

[4] The Hough clustering method used here is a variation of classical ones for finding an object's pose. A Hough transform accumulates evidence for coordinate transformations in a parameter space whose axes are the quantized transformation parameters. Large clusters of similar transformations in that space are taken as evidence of a correct match. For details please refer to [37].

The results show that the execution time grows rapidly with the complexity of each model, as well as the number of total models. The performance of the algorithm depends on the error in the measurement of sensed data points (such as positional error of points and angular error of surface normals). Partial symmetries in the objects often lead to multiple interpretations and account for increased matching time and most of the matching errors. Additional constraints from a careful symmetry analysis may resolve the ambiguities. The interpretation is also sensitive to the sensing strategy such as the size of spacing between grid points. Exploring sensing strategy to disambiguate multiple interpretations is considered in [32].

2. *Local constraints:*

   RAF uses only local geometric constraints. This is in line with the assumption that only sparse data are available through sensing like tactile sensors. Simple constraints, such as distance, angle, and direction constraints, are proven to be effective for isolated objects. However, overlapping objects increase the complexity dramatically. Other type of constraints can potentially be used if they can be obtained or derived readily from measurements. Grimson [36, 39] gives a formal combinatorial analysis of the complexity of recognition algorithm based on searching the interpretation tree, under some probabilistic assumptions on the data distribution. He shows that the expected complexity is quadratic in the number of features for isolated objects, and can be reduced to polynomial time in cluttered environments using Hough transforms and premature termination.

   It is interesting to notice that more complicated constraints, like the coupled constraints discussed in [33], do not significantly reduce the size of search, and instead increase the search time. Two reasons account for the result: the coupled constraints do not offer more information than simple constraints and are more expensive to compute. Introduction of new features and constraints should take into considerations the computational aspects of features such as how easily and reliably they can be computed.

3. *Heuristics:*

   The use of two heuristic techniques, the Hough transforms and premature termination, reduces the size of search. Since the Hough transforms focus matches on "good" candidates, the branching factor of the interpretation tree is cut down. The sensitivity of Hough transforms is discussed in a later paper by Grimson [37]. The premature termination guides the search to maximize some measure, in this case the sum of areas of faces explored in an interpretation, and prematurely terminates the search at some threshold. The use of the heuristic of maximizing area is a patch for the complexity explosion due to the introduction of null nodes. The premature termination imposes a limit on the height of the interpretation tree by thresholding and therefore reduces the length of the search paths. However, a search with bad thresholding can miss the best interpretations or accept a "good" interpretation when a better one exists. The tradeoff here is the efficiency versus quality of interpretation. Thus the choice of appropriate measure for matching quality and appropriate threshold to terminate the search is the key. Grimson [38] shows that the optimal threshold can be found, again under some assumptions on the data distribution.

4. *Further improvement:*

   In addition to what we have discussed above, the performance of RAF can be further enhanced by

   - *Grouping:* Initial grouping of features, such as edges, that likely come from same object can further reduce the size of the search and improves the accuracy and robustness of the interpretation. Jacobs [64] has shown that speedup of several order of magnitude can be expected, by using some simple grouping techniques.
   - *Search attention focus:* Techniques similar to the Hough transforms can be used to filter candidates for matches so that the "seeds" left can direct and constrain the search. More distinctive features can also be used, if we can find them, to focus matches on a smaller set of features.
   - *Use of more model knowledge to constrain the search:* Additional knowledge of the models, for example global features, can be used to further constrain the search. At the hypotheses generation stage more model knowledge can be assumed. Incorrect hypotheses can be rejected at the verification stage.
   - *Hierarchical representation with indexing:*

     For objects that are curved, the number of edge segments used to approximate the objects is large. Hierarchical representation such as a "strip tree" can be used [28].
   - *Iterative transformation computation:*

     RAF uses a few data points to estimate the model to scene transformation. An average over many individually computed transformations is used to reduce the effect of errors. A better method would iteratively improve the estimate of the transformation when more data are available. The initial estimate of the transformation can even be used to prune the tree search at the hypothesizing stage. Therefore the hypothesizing stage is interleaved with the verification stage. At each iteration features are predicted, subject to the local constraints such as those of RAF. A partial verification checks the prediction using the estimate of the transformation. The result of the partial verification is then used to refine the estimate of the transformation. However, presently rotations in RAF are represented in terms of angle/axis and orthonormal matrices that do not lead to a simple iterative refinement. Recomputing the transformation at each iteration would be too expensive. The quaternion representation for rotations of Faugeras [25] is an elegant alternative that leads to an iterative least square method.

3.3 Ikeuchi et al.

3.3.1 Overview

The approach of Ikeuchi et al. at CMU [60] deals with automatic generation of object recognition programs. It is one of a few model-based recognition systems that model sensors, in addition to geometries of objects. Ikeuchi et al. discussed issues and techniques for automatic generation of recognition programs by compilation, based on a method in an earlier paper [59]. To evaluate the approach of Ikeuchi et al., we ask the following questions: (1) what is its intended domain? (2) how well does it work? (3) how can the method be generalized to handle more complex situations? More specifically, we discuss how their method models objects and sensors and how noise is treated.

The system of Ikeuchi et al. works in bin-picking task domains, using 3D information including depth maps. It assumes that only instances of a single object model, possible jumbled, are present in the image. It does not handle occlusions, since it assumes that it can always pick the top object first. The task of the system is to automatically generate recognition programs that determine the precise position and orientation of an object at the compile time. If general model-based recognition consists of hypothesizing and verification (model test), then this system only handles the model test part that produces the transformation. On the other hand, Ikeuchi et al. present a complete methodology of programming for vision recognition.

Historically edge-based approaches have attacked bin-picking problems by focusing on brightness changes and working on the resulting binary images. These algorithms work fine for 2D objects on a flat table with a well defined background. However recognition of more complicated scenes containing 3D objects requires more knowledge of the image and the objects. Ikeuchi et al. treat the task as being extraction of useful features and control of using these features to recognize objects. They present a method for automatically compiling object and sensor models into recognition programs, based on a careful analysis of the object models, the sensor models, and the characteristics of their interplay in the scene.

The system of Ikeuchi et al. consists of the following key steps:

1. object modeling — geometric and photometric properties;
2. sensor modeling — sensor characteristics and variations of feature values in terms of detectability and reliability;
3. prediction of appearances — aspects;
4. strategy generation — interpretation tree;
5. program generation — object-oriented programming.

The object model is assumed to be polyhedral. First all the possible appearances of the object are enumerated. Ikeuchi et al. use aspects and features associated with each aspect as intermediate representations. An aspect is defined as a topologically equivalent class of appearances of an object. Shape changes between aspects, the aspect changes, are nonlinear. Shape changes within an aspect are called linear changes. All the appearances of an object model are classified into different aspects. The expected values of features for each aspect are predicted and used to discriminate different aspects. The linear change is then determined by a transformation from the aspect of the object to the appearance of the object.

An interpretation tree classifies an appearance of an object into one of its aspects that correspond to the leaves of the tree and determines the linear change within the aspect. Figure 5 shows an interpretation tree generated for a simple industrial part. The root of the tree represents all the aspects of the object while the leaves are five distinct aspects of the object, S1, S2, S3, S4, and S5. Features such as inertia and local geometries are used to discriminate aspects at each node. Within each aspect class, i.e., within each leaf node, linear changes are then determined using the EGI[5] and other features in the intermediate coordinate systems — the face and edge feature coordinate systems. Thus the interpretation tree represents a recognition strategy. Applying an interpretation tree to an appearance of the object in the scene classifies the appearance into one of the aspects and then determines its precise position and orientation within the aspect.

The aspects and their predicted features are view-dependent. The runtime efficiency is improved by enumerating all the possible variations of features due to variations in viewing directions, for example aspects, and predetermining the search path in the interpretation tree. Sensors and objects are explicitly modeled. Features are quantitatively modeled in terms of detectability and reliability[6], as opposed to other recognition systems, for example RAF, that only use "hard numbers"— step intervals—to model uncertainties of features. The feature values for the scene object are obtained from three maps, the needle map (for surface normals), the edge map, and the depth map. Dual photometric stereo can be used to produce the maps.

The interpretation tree is then compiled into an executable program. The compiled special-purpose recognition program is for a single object model only and can be used to classify instances of the object in applications such as bin-picking tasks. Object and sensor modeling and analysis of the best recognition strategy are done at compile time. This partially alleviates the repeated development efforts in the traditional one system/one application practice.

[5] The EGI (Extended Gaussian Image) can be used to achieve certain tasks such as constraining the viewing directions and rotations at the stage of determining linear change. It has the advantage of being invariant under translation and scaling of axes of the coordinate system. It also rotates in the same way as the object it represents. However, it is less powerful in cases of occlusions [46, 59].

[6] Sensor detectability specifies under what condition a sensor can detect a feature, while sensor reliability is a measure of confidence for the detected features over the detectable configurations.

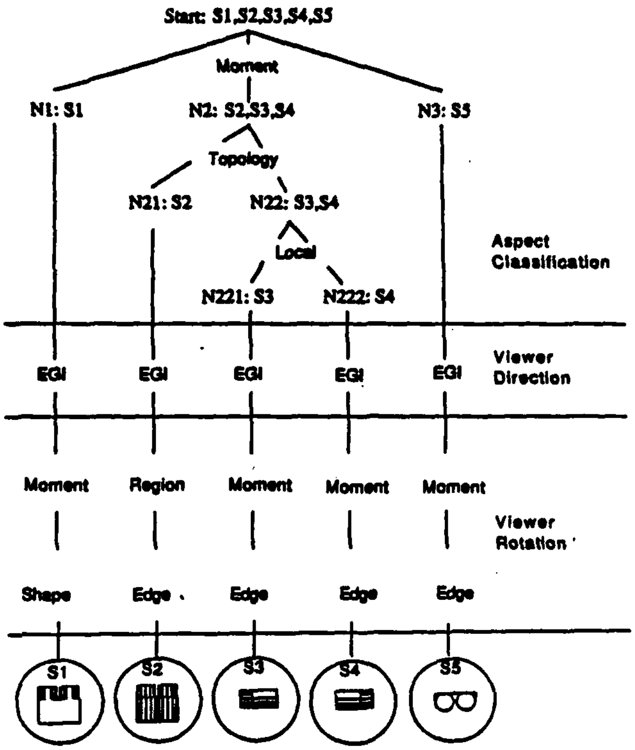

Figure 5: Interpretation Tree (from Ikeuchi et al.)

When applying the recognition program to a scene, the highest object instance of the scene is first chosen. This is designed to avoid complications due to partial occlusions by nearby instances. After the localization of the highest object, the object is picked up from the bin. Recognition proceeds to the next highest object. This sequence of recognition and action is natural of bin-picking task.

3.3.2 Evaluation

We can view the approach as being composed of two steps: classifying nonlinear shape changes (aspect changes) and determining linear shape changes within an aspect. In the first step the search is guided by focusing on aspects only at the coarser level. The classified aspects then serve as an intermediate representation and relatively cheap determination of linear changes proceed. Ikeuchi et al. consider only three degrees of freedom in the linear change: two degree of freedom in translation on the image plane and one degree of freedom in rotation around the optical axis. The camera model assumes orthographic projection. Since we have already got a rough estimate on the orientation of the object by classifying aspect changes, the rotations of edges within an aspect are more sensitive around the viewing direction than around axes within the image plane. However, we still need the depth information for picking the object from the bin. A depth map is used for resolving this problem.

Although the scheme works well for the task at hand, bin-picking of relatively simple industrial parts, it does not easily generalize to more general tasks that handle realistic scenes, aside from the techniques for automatically compiling recognition programs. The system assumes that the linear changes within an aspect can be determined cheaply and easily. This assumption restricts the class of objects it can handle, that is, polyhedral objects. The polyhedral approximations of smooth objects are not stable. Its expected values of features are sensitive to small variations in viewing direction. For instance, the polygonal approximation to a circle is sensitive to small variations in the image. A slight rotation of the approximating polygon results in drastic errors in determining the transformation [34]. Most real-world objects display complex shapes whose nonlinear shape changes can not be easily classified into aspects. Although polyhedral approximation to curved objects can sometimes be used with the aid of feature reliability, it nevertheless introduces matching errors.

The tradeoff between runtime efficiency and speed and storage space is evident in Ikeuchi et al's decision of precompiling the model and sensor features and the recognition strategy into an executable program and merely applying the program at runtime, as opposed to invoking the features and constraints and the analysis of how to use them at runtime. The size of the interpretation tree can potentially be very large. In the worst case, the size is exponential in the number of distinct aspects of the object model, which can be large for objects that lack planar faces. Therefore judicious choice of features is of importance. A cutoff in the enumeration of appearances is used, with some sacrifice in accuracy.

Precompiling constraints into a tree also predetermines the search path. The method is sensitive to small variations in the geometries of objects and sensors, for example defects in objects due to manufacturing processes. Although Ikeuchi et al. model some of the effect with feature reliability, this trades off some runtime flexibility. For bin-picking tasks starting recognition at the highest instance does reduce the possibility of part of it occluded by other instances. Future research should generalize this method to handle occlusions due to different objects in general scenes. Backtracking of the interpretation tree search is needed to handle the occlusions.

3.4 Related work

This section surveys related work to the three approaches we have just discussed. One group deals with line drawing interpretation where the paradigm of consistent labeling has its roots. The others either are similar to the three approaches or represent slightly different approaches in the area of model-based recognition.

3.4.1 Interpreting line drawings:

1. *Guzman:*

   Guzman [41] started the systematic work on interpreting polyhedral line drawings. The task is to partition a picture of line drawings into distinct bodies of objects. The pictures are assumed to contain polyhedral objects with trihedral vertices only. The reason for choosing the polyhedral world was that it possesses some essential aspects of scene analysis (image formation, models, spatial relationship of scene objects — occlusion, shadowing, support, etc.), and yet is simple enough to admit computationally tractable analysis at the time (around the 60's and early 70's) [87]. The approach presents a theory about how to place links on pairs of two regions that belong to a single object body, based on an analysis of junction formations. Given a picture, first it labels local pairs of regions according to the theory. Then the local evidence is grouped to produce a legal interpretation. This works well on many examples; but it fails on some simple scenes. To patch up the hole, additional heuristic rules are added. The approach claims to use no knowledge of prototypical bodies. The basic flaw of Guzman's method is that it was necessary to add and modify rules to handle counterexamples.

2. *Huffman & Clowes:*

   Huffman and Clowes [49, 20] suggested that the remedy to Guzman's scheme comes from a complete analysis of the junction types. Edges forming a junction can be labeled as one of convex, concave, or occluding edges. Since there are only a small number of legal labelings for each junction type, all the legal labelings of junctions can be enumerated to form a catalogue of legal labels. Each edge bridges two junctions, if both of them are visible. The two labels at the ends of the edge have to be consistent. Interpretation is merely a tree search that backtracks when it comes to a label that is inconsistent with what is already known. In a later paper Huffman [51] generalizes the idea to handle polyhedral objects with arbitrary number of edges forming a junction, using decision criteria in a dual space to check for legality of a label.

3. *Waltz:*

   Waltz [85] extended the semantic catalogue of Huffman/Clowes to handle shadows. He presented an algorithm that first locally eliminates all the illegal labels at each junction by checking pairwise consistency across each edge, and then proceeds to do a global search. The remarkable thing about Waltz's algorithm is that the filtering pass (the first step) eliminates almost all the impossible interpretations and achieves drastic speedup. The approach relies on two basic components: a catalogue of semantic labels and combination rules and a single mechanism — constraint propagation — that uses the catalogue for consistent labeling. He thus showed that the interpretation of a scene is easy if we can come up with a dictionary of primitives — "words" — and a grammar to put them together to produce legal "sentences".

The Waltz algorithm shows that the more we know about the world, the smaller the search space is. If we can discover and represent all the intrinsic constraints of some complex objects, we should be able to apply the algorithm in much the same way as in the polyhedral world. For example, curved objects offer rich information such as local curvatures. Careful analysis should lead to a successful program to handle general (smooth) surfaces [18]. Brady [10] also presents a primal surface sketch to describe significant changes such as steps, roofs, and ridges.

4. *Other extensions:*

   Later research has extended Waltz's work in several directions: Sugihara [79] looks for new sources of constraints such as coplanarity of vertices and edges that lie in the same face of the polyhedral object. Mackworth [66] and Draper [22] work in the gradient space and explore positions and orientations of edges and surfaces. Chakravarty [18] and Malik [68] generalized the method to handle curved objects.

5. *Remarks:*

   It has been shown that there are picture configurations that are locally consistent but globally inconsistent [49]. The reason seems to be that the local constraints of corner sorts are incapable of expressing global constraints [67]. One way to overcome this is to explore stronger local constraints. Nguyes [74] showed that by exploring local topological constraints local consistency implies global consistency. Another method is to explore some degree of global constraints. The knowledge of models can be used to eliminate globally inconsistent interpretations. This naturally leads to model-based recognition that hypothesizes a match to a model and then verifies the match.

   One important lesson we learned from the work on line drawing interpretation is that in order to improve the interpretation of a scene we should first search for better descriptions of the scene and model features, and then look for more evidence. Throwing in ad hoc heuristics to deal with counterexamples should only be a last resort [87].

3.4.2 Searching a tree to match model and image features

1. *Faugeras & Hebert:*

   Faugeras and Hebert [25] presented a 3-D recognition and positioning algorithm using geometric matching between primitive surfaces. The method (1) estimates rotation and translation (when the transformation admits decomposition) by pairing primitive surfaces to those of models, and then (2) iteratively refines the estimate using a least square method that minimizes a measure of matching quality. Rotation is represented as quaternions. Recognition is a tree search procedure. The search is pruned by a local consistency measure that precludes strongly inconsistent pairs from consideration.

2. *HYPER*[7] *(Ayache & Faugeras):*

   HYPER [26] identifies and locates objects by generating and verifying hypotheses coupled with a recursive estimation of the model to scene transformation. It handles 2D objects on a flat surface and is robust to partial occlusions, shadows, and scalings. The system has been successfully tested in a large number of different scenes containing partially overlapping industrial parts. Local and compact descriptions for describing both the models and scenes are used. Hypotheses are generated to estimate the model to scene transformation by matching a few privileged segments of descriptions to those in the scene and are ranked according to a measure of quality. Only the best hypotheses are evaluated. The transformation is then evaluated by matching additional segments and recursively updating parameters of the transformation using a least square method (Kalman filter). The privileged segments of descriptions are served to focus search onto a small number of distinct features. The hypothesized match constrains the search relative to the model. This is similar to that of the alignment method [83].

3. *Local-feature-focus Method:*

   This method [7] uses local features such as regions and corners of 2D parts to direct the generation of hypotheses. It finds one feature, the focus feature, and uses it to predict a few nearby features to look for. A graph-matching technique is used to identify the largest cluster of image features matching a cluster of object model features. The matching is solved as a maximal-clique problem, which is very complex. However, the local-feature-focus method prunes the tree search by cutting down the branching factor of the tree. The key to this method is to select the best focus features and the most useful nearby features, through the ranking of features, so that only the best hypotheses are generated and the time for (expensive) verification step is minimized. The selection of focus features is done automatically.

4. *3DPO:*

   The task of 3DPO [8] is the bin-picking with 3D objects, possibly occluded. Range data are available as input. This method generates hypotheses and matches them against image features. Search is pruned by a consistency of features. A few features or cluster of features are automatically selected from a CAD model based on a feature representation and preliminary planning. The image is clustered into view-independent clusters of features. Hypotheses are generated based on the features detected, possibly with the help of the local-feature-focus method.

5. *Grouping of features(Lowe and Jacobs):*

   Lowe [65] presented a method to recognize 3D objects from 2D images. The method (1) forms groupings and structures in images; (2) uses a probabilistic ranking to reduce the size of search during matching; and (3) matches spatially 2D image and

[7] HYPER stands for HYpotheses Predicted and Evaluated Recursively.

model by determining a transformation. The groups in the image bridge the gap between the 2D images and 3D models.

Jacobs [64] showed that initial grouping of features, such as edges, that likely come from same object can reduce the size of the search and improves the accuracy of interpretations. He uses two types of geometric constraints to form grouping of edges, the distance between two edges and their relative orientation. The method achieves a reductions in computations of between a factor of 100 and 1500 over an identical system that does not use grouping.

6. *Ikeuchi & Horn:*

   Ikeuchi, Horn, et al. [57] solved the bin-picking of 3D objects by matching the surface normal distribution of image object against that of the model. The surface normal distribution of the scene is calculated by the photometric stereo method. The scene is then segmented into isolated regions using the surface normal distribution. Finally the method determines the object orientation by matching the surface normal distribution of the target region against that of the known object model using the discrete Extended Gaussian Image.

7. *Baird:*

   Baird [1] equates consistent matches with constraints in some geometric space and uses a linear programming method to solve this geometric problem.

3.4.3 Using models to filter matches

Alignment Method:

The alignment method [83] searches directly for possible transformation from a model to image object using as small a number of points as possible and discards incorrect transformation by checking with additional points. This is another instance of hypothesize-and-verify paradigm. It recognizes an observed object by hypothesizing the object's identity together with its position and orientation and verifying the hypothesis.

The alignment method first aligns an image object with a model by a transformation. Three distinct points or surface normals are used to determine the transformation. For each of the models in the model library, a transformation to the image object is determined. Then the fitness of all transformations is examined and ranked according to some metric cf quality, for example the simple distance metric. The best fit is the correct interpretation of the observed object. It is helpful to compare the alignment method with Grimson et al.'s RAF which tries to find all the correct interpretations—a decision that is purely driven by the task to accomplish. The alignment has been generalized to handle nonrigid transformations. Further extensions are carried out by Basri and Ullman to recognition of objects with smooth surfaces using a curvature model, and recognition based on linear combination of models [81, 2].

3.4.4 Automatic generation of recognition programs

Goad:

Goad [29] describes a method for automatically constructing special purpose recognition program. The method is based on precomputing tables containing bounds on spatial relations between selected pairs of image features for small range of viewpoints. A search for matches between image edges and model edges is made. Hypotheses are checked by a table lookup. The method is claimed to be fast.

4 A Perspective on Model-based Recognition

We have evaluated the three approaches of ACRONYM, RAF, and Ikeuchi et al. in the previous section and have discussed the strengths and weaknesses of each approach. We have also suggested future directions for improvements in their performance.

In this section we will discuss and compare the three systems in the larger context of other approaches to model-based recognition, and place them in the comparative framework defined by the three axes: generality, representation, and control, as shown in Figure 6. We will identify where tradeoffs are made and analyze why they are made.

The following types of tradeoffs made in the implementations will be discussed to reveal how the systems' tasks affect the decisions underlying the tradeoffs:

- efficiency vs. generality;
- efficiency vs. accuracy;
- efficiency vs. runtime flexibility;
- efficiency vs. storage space.

From the three systems under study and other work we can see a constant battle to achieve certain performance without too much sacrifice in generality. The issues of representation and control have been raised again and again. The ACRONYM system is the most complex system. It is one of a few implemented recognition systems that are intended to be general and domain-independent and has been tested on some aerial images. However, it would be difficult for the general constraint manipulation system to compute constraints of object parameters from an unconstrained viewpoint. ACRONYM further constrains the parameter ranges of models by restricting the range of viewpoints, when it is applied to interpret real images, for example the aerial images of airfields. Binford noted in [6], "there is no profound reason why ACRONYM could not recognize aircraft in images taken at ground level, although it will probably break when tested on

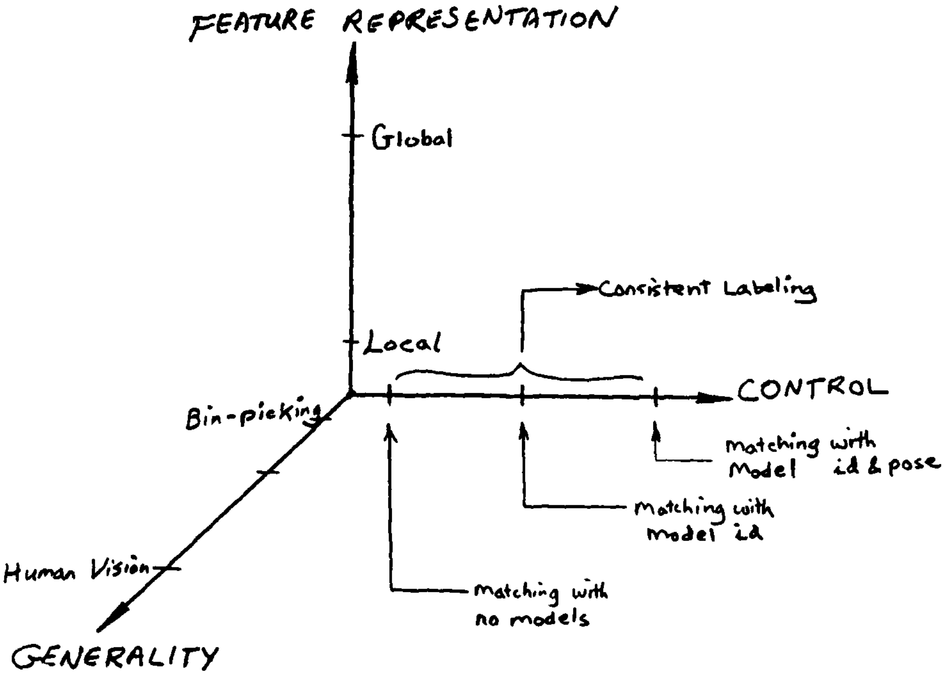

Figure 6: The comparative framework for different approaches

such images because of bugs or missing capabilities that were not exercised previously." Although ACRONYM is intended to be a general system, in order to work on real images many tradeoffs have been made along the way from its general theory to the working implementation. The complexity of the implementation of ACRONYM makes it very hard to test other ideas that the designers had in mind. This fact motivates the need for reasonable computational efficiency so that various ideas of a design can be tested and improved afterwards. Waltz's algorithm, for example, is very fast in interpreting line drawings. Waltz [85] noted that because of the speed of his algorithm he was "able to test the program on many separate line drawings" and had "been able to gain a clearer understanding of the capabilities and ultimate limitations of the program."

RAF improved performance by specialization of domains. RAF works in a robot sensing environment where accuracy and speed are the main objectives. Its input consists of a sparse set of data including positions and surface normals which generates a smaller interpretation tree than that of dense data. RAF tends to generate all the possible interpretations on the belief that false negatives are better than false positives (false positives are hard to get rid of).

Ikeuchi et al. presented a method to automatically generate a recognition strategy and compile it into a program. Although the principles of their system are general, their ability to work out their theory relies on the fact that appearances of an object can be enumerated. This is true for simple industrial parts in most of bin-picking tasks. The efficiency of the method comes from the precompilation into a table of constraints and their use in recognition, based on a decomposition of transformations into nonlinear changes and linear changes. More complicated objects do not lend themselves to such a clean separation of changes in appearances. It will be an interesting research issue to see how this method handles multiple models and occlusions.

RAF is specialized by exploring model specific knowledge at compile time. Its constraints are all precompiled into tables so that at the runtime only table lookup is used to obtain constraints. It is fast. In contrast, ACRONYM uses a general constraint manipulation system to reason about constraints at runtime. As a result ACRONYM is more robust but is slower. Ikeuchi et al.'s approach goes one step further by precompiling into a table the constraints *and* the use of the constraints in discriminating hypotheses of object attitude.

Both ACRONYM and RAF use coordinate-frame independent, i.e., sensing geometry independent, representations for constraints. In ACRONYM, object shapes are hierarchically decomposed into parts which are linked together by constraints in terms of algebraic inequalities. The constraints of RAF characterize local geometry of object shapes. Ikeuchi et al.'s method, however, adopts a viewer-centered representation (i.e., aspect representation) by enumerating all distinct aspects of an object. Errors are explicitly modeled. In general, the constraints should model errors explicitly and be coordinate-frame free, in addition to being easy to compute and robust to noise [40]. As a result, the constraints are invariant to viewing directions. A quantitative estimate of feature detectability and reliability can be obtained.

The extent of model knowledge used in controlling the search varies among different approaches. Ullman's alignment method aligns a model with the observed object by a few points and uses the model to constrain the match and verification afterwards. The constraints are thus global. 3DPO uses a few distinctive features to find the transformation and verify it with additional features. Therefore the search is directed to distinct features at the beginning of the search. The few distinctive features are selected based on the knowledge of models. RAF uses only local geometric constraints to prune the search. Ikeuchi et al. use the model appearances to determine transformation.

Globally computing transformations from models to scene is more expensive than locally checking for consistency between respective model features and image features. It is not surprise to see that the alignment method is somewhat slower than RAF [40]. Preliminary grouping of features into extended features which provide stronger constraints will improve the quality of hypotheses. Model verification is therefore cheaper. RAF uses the extended features (edges) to reduce the need for search. The use of edges, as opposed to points, cuts down both the branching factor and the height of the interpretation tree. ACRONYM uses an edge-linking algorithm to group edge segments together.

The nature of constraints used in ACRONYM and RAF are local geometric constraints. Ikeuchi et al. use global properties such as moments and the EGI (Extended Gaussian Image) and as well local properties like edges and regions.

Global features, such as area, perimeter, moments, and elongation commonly used in classical pattern recognition task for industrial parts, are sensitive to occlusions and illuminations. They are also more expensive to obtain. However they are less susceptible to variations in object local parameters. In many cases, global feature offer tighter constraints on the search. Grimson [30] showed that using the constraint of more distant pairs of points at the beginning of the tree search most effectively constrains the size of the interpretation tree generated. On the other hand, local features are invariant to global changes such as occlusions and shadows, at the expense of being weak in constraints. If recognition of an object is from a known viewpoint, then the appearance of object is easy to predict (computationally easy and cheap). Some global features and constraints can be accurately predicted. If recognition of object is from an unknown viewpoint, then, unless we have ways of enumerating all the possible view directions (as Ikeuchi et al.'s use of aspects), local constraints are more applicable.

Dual space representations of images make the implicit properties of the original images explicit. Examples of dual spaces include the gradient space representation of points, edges, and faces [50] and the EGI (Extended Gaussian Image) [46] for surface orientations. As a result, certain type of analysis is very easy in the dual spaces. It is interesting to note that some dual space representations can convert global features into local ones. For example, the global periodicity of time signals are local in the frequency domain through Fourier transform.

5 Towards General and Robust Recognition Systems

5.1 Versatile recognition systems

A versatile recognition system has to address the issues of handling a broad range of tasks and integrating multiple visual cues. This requires that the system employ mostly view-independent and task-domain-independent modeling process and is robust to occlusions and shadows in scenes and noise in measurements. To deal with complex scenes the system should use whatever available cues, such as edge, color, shading, texture, motion, and stereo, to name a few, to accomplish the task.

We will discuss the steps towards a general and robust recognition machine, based on some of the machinery developed so far.

5.2 Better modeling and representations

5.2.1 Modeling and representing realistic scenes

Many systems do not generalize to deal with occlusions, poor lighting, and shadows. Some are susceptible to noise in measurements and variations in sizes of objects. ACRONYM uses volumetric primitives, the generalized cones, to represent objects. RAF and Ikeuchi et al. approximate the world with polyhedrons. Ikeuchi et al. also use feature reliability to discuss the crudeness of the approximation.

To accomplish many difficult tasks, objects and sensors in general should be modeled and represented explicitly to reveal their intrinsic properties and expose their constraints both qualitatively and quantitatively. Presently in many applications, models are usually specified in terms of geometric properties only, partially because of the tasks to accomplish and the availability of sensed information. Study on image formation may lead to discoveries of other constraints on objects. Horn [48] has investigated the problem of recovering shape from shading using image brightness cues. Ikeuchi et al. have taken an important step towards sensor modeling [62]. More recently, fractal representation for natural forms has been suggested [75]. Much research remains to be done.

Most systems assume objects to be rigid. However many real world objects, like moving bodies of animals, are nonrigid. Techniques for recognition of nonrigid objects need to be developed. Grimson [35] and Ullman [83] have attempted the problem.

The use of class/subclass relations in ACRONYM is an interesting approach to handle generic classification. ACRONYM does not commit itself to make a particular classification unless it has enough information to do so. Some tasks, such as recognizing a car as a generic object consisting of a frame mounted on four wheels, as opposed to a specific car model, like a Toyota Corolla, require the recognition systems to have some knowledge of generic classes. However for objects that do not have stable and unique decompositions, more work should be done to discover what constitute their class/subclass relations.

5.2.2 Modeling errors

A robust system must realistically model measurement errors in sensory data. RAF uses an "error ball" for uncertainty in distance measurements and an "error cone" in angle measurements. This type of error models imposes uniform weighing across the error region. Ikeuchi et al. use a more realistic error model. In general, an error model should use some sort of distribution that reflects the underlying models of objects and sensors. One possibility is the Gaussian distribution [17]. Some statistical methods might be useful here. Nonuniform error modeling can be expected to improve the accuracy and reduce the need for search.

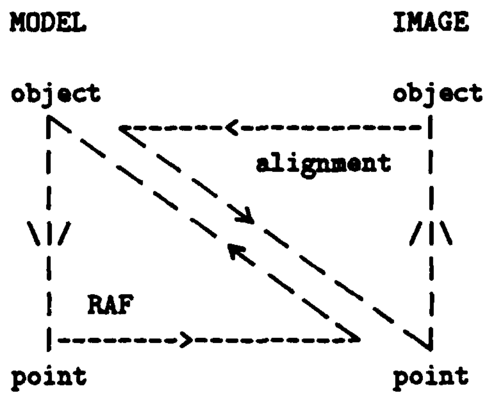

Figure 7: The use of model knowledge for hypothesize-and-verify in RAF and the Alignment Method.

5.3 Further reduction on search

5.3.1 Use of model knowledge

As we noted previously, recognition can be viewed as solving two problems [31]: *what* is in the scene and *where* the object is. RAF solves *what* by hypothesizing the identity of an observed object and then determines *where* by verifying the hypothesis. At the step of *what* only local constraints of the model is used. The verification (model test) is done after *what* is solved. The alignment method solves *where* before *what*. In this sense alignment method uses *where* (performing the model test right at the beginning) to constrain the search for *what*.

The extent to which knowledge of models is used in generating the hypotheses, hypothesizing the identity of an image object, or predicting its 3D position and orientation as well, depends on the cost of computing a transformation relative to that of exhaustive matching of features. Figure 7 compares RAF with the Alignment Method. It shows at which level the matches occur, and how much knowledge of the models is hypothesized.

5.3.2 Intermediate representations and search focus

In a cluttered world of complicated objects recognition soon becomes computationally intractable. The search space for all the matches to be considered can be enormous. The size of search can be reduced in two directions:

1. using intermediate representations (e.g. grouping of edges);

2. focusing attention of search;

If we view recognition as a mapping from the sensed data to labeled objects, then the search for the correct matches is to develop a search tree (in [33] the interpretation tree). It is too expensive to conduct a brute-force matching. Intermediate representations are needed to bridge the semantic distance between the input and the output of the recognition system.

Dense features can be grouped into extended features according to some similarity measure (for example [64]). The matches are then carried out on the intermediate representation, the extended features. The grouping of features cuts the original search tree into a set of shallower trees. The problem here is to choose the criteria for grouping. The grouping techniques of [65, 64] are successful attempts in this direction. It is shown that the initial grouping of features reduces the size of search; more specifically, it cuts down both the height and the branching factor of the interpretation tree, and increases the robustness of the system to small variations in object and image parameters. Yip [88] used intermediate representations of the phase space of dynamical systems to bridge the input with the output in the area of automatically recognizing Hamiltonian systems.

Alternatively, the original set of dense features can be subdivided into several subsets. Each of the subsets is then matched to the model. The first successful match constrains *quantitatively* the matches on the rest of subsets. The branching factor of the search tree is cut down afterwards. What subset to try first affects the performance. Finding distinctive features [8], such as locations of holes and corners, and matching them against the model falls somewhat into the scheme of feature subdivision. Alignment method is another instance of the subdivision [83]. The difficulty is to decide what constitute distinctive features. More work should be done on how to find and extract distinct features reliably.

Putting the most effective constraints at the beginning of the search essentially constrains the branching of the search tree. How to rank constraints with their effectiveness, especially when they come from different sources, remains an open research problem. The answer will be some metric of quality for constraints.

5.3.3 Remark

Since we have a fairly good understanding of the effectiveness of simple geometric constraints, more work should be directed towards (1) the discovery of better constraints; (2) the integration of constraints from multiple visual cues; and (3) the study of their combined effects. To achieve performance comparable with human beings, using a library of techniques we have developed and invoking them for appropriate situations at appropriate time is key.

6 Conclusion

The principal issues in machine recognition — generality, representation, and control — have been identified and their principles and requirements have been motivated. We have presented a comparative framework for the evaluation of different approaches to machine recognition, particularly those of ACRONYM, RAF, and Ikeuchi et al., and have discussed and compared these approaches with respect to the three issues: generality, representation, and control. These different approaches to machine recognition constitute an important class of recognition methods, namely model-based recognition that has been successful compared with its counterparts. This paradigm formulates recognition as the correspondence between the image objects and model objects and searches for matches that give the correct interpretations to the image.

Various tradeoffs made in the implementations, such as efficiency vs. generality, efficiency vs. accuracy, and efficiency vs. runtime flexibility, have been identified and analyzed with respect to the systems' generality and intended performance. We conclude that the decisions underlying these tradeoffs are mainly task-driven.

We have formulated the principles and requirements of representation and control for machine recognition. The issues of usefulness, computability, invariance, and robustness of representations have been addressed in the context of ACRONYM, RAF, and Ikeuchi et al., along with other work. We have discussed the common control strategy used in these approaches: hypothesize-and-verify. The strategy hypothesizes a match between the image object and the model and verifies that the match actually produces a legal interpretation of the image. We have pointed out the strengths and weaknesses of each approach with respect to its modeling of objects, its robustness to errors, its control of search, and the extent of model knowledge used in constraining the search for consistent matches, and suggested future improvements for each approach.

We have discussed the steps towards general and robust recognition systems. Based on our analysis on the use of feature grouping and search focus techniques in reducing the size of search, we have suggested the intermediate representations to bridge the gap between the object models and the images. We argue that more realistic modeling and representation of objects and errors and more efficient control strategy is the key to versatile recognition systems.

7 Acknowledgments

This paper is a revised version of the area exam report. I would like to thank Gerry Sussman and Berthold Horn for the support and encouragement; Eric Grimson, Rod Brooks, and Shimon Ullman for helpful discussions; Franklyn Turbak, David Jacobs, Mike Eisenberg for debugging my thoughts and improving the presentation tremendously; and

Tao Alter, David Clemens, David Beymer, Ken Yip, and Jim O'Toole for discussions. This work would not be possible without the understanding and encouragement from Ying Yin.

References

[1] H. S. Baird, "Model-Based Image Matching Using Location." *PhD thesis,* Princeton Univ., Oct. 1984.

[2] R. Basri and S. Ullman, "Recognition by Linear Combinations of Models." June 1989.

[3] Paul. J. Besl and Remesh C. Jain, "Three-Dimensional Object Recognition." *Computing Surveys,* 17(1), March 1985.

[4] Paul Besl and Remesh Jain, "Intrinsic and Extrinsic Surface Characteristics." *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Francisco, June 1985.

[5] Thomas O. Binford, "Visual Perception by Computers." *IEEE Syst., Sci., Cybern. Conf.,* Miami, invited paper, Dec. 1971.

[6] Thomas O. Binford, "Survey of Model-based Image Analysis Systems." *The International Journal of Robotics Research,* 1(1), Spring 1982.

[7] R. C. Bolles and R. A. Cain, "Recognition and Locating Partially Visible Objects: the Local-Feature-Focus Method." *The International Journal of Robotics,* 1(3):57-82, Fall 1982.

[8] R. C. Bolles, P. Horaud, and M. J. Hannah, "3DPO: A Three-Dimensional Part Orientation System." *Proc. 8th International Joint Conf. on Artificial Intelligence,* West Germany, August, 1983.

[9] M. Brady (ed.), *Computer Vision.* North-Holland, 1981.

[10] M. Brady, J. Pence, and A. Yuille, "Describing Surfaces." *The 2nd International Symposium on Robotics Research,* 1985.

[11] Rod Brooks, "Goal-Directed Edge-Linking and Ribbon Finding." *Proc. DARPA Image Understanding Workshop,* Pala Alto, California, April 1979.

[12] Rod Brooks, "Geometric Reasoning in ACRONYM." *Proc. DARPA Image Understanding Workshop,* Pala Alto, California, April 1979.

[13] Rod Brooks, "The ACRONYM Model-Based Vision System." *Proc. IJCAI-79,* Tokyo, August 1979.

[14] Rod Brooks, "Symbolic Reasoning Among 3-D Models and 2-D Images" *Artificial Intelligence* 17, pp. 285-348, 1981.

[15] Rod Brooks, *Model-Based Computer Vision.* UMI Research Press, Ann Arbor, Michigan, 1981.

[16] Rod Brooks "Model-based Three-dimensional Interpretations of Two-Dimensional Images" *IEEE PAMI,* 5(2):144-149, 1983.

[17] Rod Brooks, private communication.

[18] I. Chakravarty, "A Generalized Line and Junction Labeling Scheme with Applications to Scene Analysis." *IEEE PAMI,* 1(2), April 1979.

[19] Roland T. Chin and Charles R. Dyer, "Model-Based Recognition in Robot Vision." *Computing Surveys,* 18 (1), March 1986.

[20] M. B. Clowes, "On Seeing Things." *Artificial Intelligence* 2, pp. 79-116, 1971.

[21] L. Davis, "Shape Matching Using Relaxation Techniques." *IEEE PAMI* 1, pp. 60-72, Jan. 1979.

[22] S. W. Draper, "The Use of Gradient and Dual Space in Line-Drawing Interpretation." *Artificial Intelligence,* Special Volume on Computer Vision, Vol 17, Numbers 1-3, August 1981.

[23] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[24] G. J. Ettinger, "Hierarchical Objects Recognition Using Libraries of Parameterized Model Sub-Parts." *MIT AI-TR*-963, 1987.

[25] D. D. Faugeras and M. Hebert, "A 3-D Recognition and Positioning Algorithm Using Geometric Matching Between Primitive Surfaces." *Proceedings of the 8th International Joint Conference on Artificial Intelligence,* West Germany, August 1983.

[26] N. Ayache and O. D. Faugeras, "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects." *IEEE PAMI,* 8(1):44-54, January 1986.

[27] M. Fischler and O. Firschein, "Representations and Transformations." *Readings in Computer Vision,* M. Fischler and O. Firschein (eds.), Chapter 6, pp. 645-651, 1987.

[28] Peter C. Gaston and Tomas Lozano-Perez, "Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane." *IEEE PAMI,* 6(3), May 1984.

[29] C. Goad, "Special Purpose Automatic Programming for 3D Model-Based Vision." *Proc. DARPA Image Understanding Workshop,* Arlington, Virginia, June 1983.

[30] Eric Grimson and Tomas Lozano-Perez, "Model-based Recognition and Localization From Sparse or Tactile Data." *MIT AI Memo* 738, August 1983.

[31] Eric Grimson, "Recognition and Localization of Overlapping Parts from Sparse Data." *MIT AI Memo* 841, June 1985.

[32] Eric Grimson, "Sensing Strategies for Disambiguating Among Multiple Objects in Known Poses." *MIT AI Memo* 855, August 1985.

[33] Eric Grimson and Tomas Lozano-Perez, "Localizing Overlapping parts by searching the interpretation tree." *IEEE PAMI,* 9(4), 1987.

[34] Eric Grimson, "On the Recognition of Curved Objects." *MIT AI Memo* 983, July 1987.

[35] Eric Grimson, "On the Recognition of Parameterized Objects." *MIT AI Memo* 985, October 1987.

[36] Eric Grimson, "The Combinatorics of Objects Recognition in Clutter Environments using Constrained Search." *MIT AI Memo* 1019, February 1988.

[37] Eric Grimson, "On the Sensitivity of the Hough Transform for Object Recognition." *MIT AI Memo* 1044, May 1988.

[38] Eric Grimson, "On the Verification of Hypothesized Matches in Model-Based Recognition." *MIT AI Memo* 1110, May 1989.

[39] Eric Grimson, "The Combinatorics of Heuristic Search Termination for Objects Recognition in Clutter Environments." *MIT AI Memo* 1111, May 1989.

[40] Eric Grimson, private communication.

[41] A. Guzman, "Computer Recognition of Three-dimensional Objects in a Visual Scene." *MAC-TR*-59, PhD dissertation, Project MAC, MIT, 1968.

[42] A. R. Hanson and E. M. Riseman, "Defining the Field of Computer Vision." *Computer Vision Systems,* A. R. Hanson and E. M. Riseman (eds.), Introduction, Academic Press, New York, 1978.

[43] R. M. Haralick and L. G. Shapiro, "The Consistent Labeling Problem: Part I." *IEEE PAMI,* 1(2), April 1979.

[44] R. M. Haralick and L. G. Shapiro, "The Consistent Labeling Problem: Part II" *IEEE PAMI,* 2(3), May 1980.

[45] E. Hildreth, "The Detection of Intensity Changes by Computer and Biological Vision Systems" *Computer Vision, Graphics, and Image Processing* 22, pp. 1-27, 1983.

[46] B. K. P Horn, "Extended Gaussian Images." *Proc. IEEE,* 72(12), Dec 1984.

[47] B. K. P. Horn, *Robot Vision.* MIT Press, 1986.

[48] B. K. P Horn, "Height and Gradient from Shading." *MIT AI Memo* 1105, May 1989.

[49] D. A. Huffman, "Impossible Objects as Nonsense Sentences." *Machine Intelligence* 6, B. Meltzer and D. Michie (eds.), Edingburgh University Press, Edingburgh, 1971.

[50] D. A. Huffman, "A Duality Concept for the Analysis of Polyhedral Scenes." *Machine Intelligence* 8, E. W. Blcock and D. Michie (eds.), Ellis Horward, Chichester, 1977.

[51] D. A. Huffman, "Realizable Configurations of Lines in Pictures of Polyhedra." *Machine Intelligence* 8, E. W. Blcock and D. Michie (eds.), Ellis Horward, Chichester, 1977.

[52] D. A. Huffman, "Surface Curvature and Applications of the Dual Representation." *Computer Vision Systems,* A. R. Hanson and E. M. Riseman (eds.), Introduction, Academic Press, New York, 1978.

[53] D. Huttenlocher and S. Ullman, "Recognizing Solid Objects by Alignment." *Proc. DARPA Image Understanding Workshop,* Cambridge, 1988.

[54] D. Huttenlocher, "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image." *MIT AI-TR*-1045, Oct. 1988.

[55] K. Ikeuchi, "Recognition of 3-D Objects Using the Extended Gaussian Image." *Proc. 7th International Joint Conference on Artificial Intelligence,* IJCAI-81, Canada, August 1981.

[56] K. Ikeuchi, "Determining Attitude of Object From Needle Map Using Extended Gaussian Image," *MIT AI Memo* 714, April 1983.

[57] K. Ikeuchi, B. K. P Horn, et al. "Picking Up an Object from a Pile of Objects." *MIT AI Memo* 726, May 1983.

[58] K. Ikeuchi, H. K. Nishihara, B. K. P. Horn, and A. Nagata, "Determining Grasp Configurations Using Photometric Stereo and the PRISM Binocular Stereo System." *The International Journal of Robotics Research,* 5(1), Spring 1986.

[59] K. Ikeuchi, "Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-picking Tasks" *The International Journal of Computer Vision,* pp. 145-165, 1987.

[60] Katsushi Ikeuchi and Takeo Kanade, "Towards Automatic Generation of Object Recognition Programs." *CMU-CS*-88-138, May 1988.

[61] Katsushi Ikeuchi and Ki Sang Hong, "Determining Linear Shape Change: towards automatic generation of object recognition programs." *CMU-CS*-88-188, Dec 1988.

[62] Katsushi Ikeuchi and Jean-Christophe Robert, "Modeling Sensor Detectability with VANTAGE Geometric/Sensor Modeler," *CMU-CS*-89-120, Feb 1989.

[63] D. Jacobs, "The use of Groupings in Visual Object Recognition." *MIT AI-TR*-1023, October, 1988

[64] D. Jacobs, "Grouping for Recognition." Nov. 1989.

[65] D. G. Lowe, "Three-Dimensional Object Recognition from Single Two-Dimensional Images." *Technical report* 202, Courant Institute of Mathematical Sciences, New York Univ., Feb. 1986.

[66] A. K. Mackworth, "Interpreting Pictures of Polyhedral Scenes." *Artificial Intelligence* 4, pp. 121-137, 1973.

[67] A. K. Mackworth, "How to See a Simple World: An Exegesis of Some Computer Programs for Scene Analysis." *Machine Intelligence* 8, E. W. Blcock and D. Michie (eds.), Ellis Horward, Chichester, 1977.

[68] J. Malik, "Interpreting Line Drawings of Curved Objects." *International Journal of Computer Vision,* 1(1):73-103, 1987.

[69] D. Marr, "Representing Visual Information." *MIT AI Memo* 415, May 1977

[70] D. Marr and H. K. Nishihara, "Representation and Recognition of the Spatial Organization of Three Dimensional Shapes" *MIT AI Memo* 416, May 1977.

[71] D. Marr and H. K. Nishihara, "Visual Information Processing: Artificial Intelligence and the Sensorium of Sight." *Technology Review,* 81(1):2-23, 1981

[72] D. Marr, *Vision*. W. H. Freeman and Co., New York, 1982.

[73] Lee R. Nackman, "Two-Dimensional Critical Point Configuration Graphs." *IEEE PAMI,* 6(4), July 1984.

[74] V. Nguyes, "Exploiting 2D Topology in Labeling Polyhedral Images", *Proc. 10th International Conferencce on Artificial Intelligence,* Milan, 1987.

[75] A. P. Pentland, "Perceptual Organization and the Representation of Natural From." *Artificial Intelligence,* 28:293-331, May 1986.

[76] T. Poggio and the Staff, "MIT Progress in Understanding Images." *Proc. DARPA Image Understanding Workshop,* California, May 1989.

[77] L. G. Roberts, "Machine Perception of Three-Dimensional Objects." in *Optical and Electro-optical Information Processing,* Tippet et al. (eds.), MIT Press, 1966.

[78] Azriel Rosenfeld, "Computer Vision: Basic Principles." *Proc. IEEE,* 76(8), August 1988.

[79] K. Sugihara, *Machine Interpretation of Line Drawings.* MIT Press, 1986.

[80] D. W. Thompson and J. L Mundy, "Three-Dimensional Model Matching from an Unconstrained Viewpoint." *Proc. IEEE Conf. on Robotics and Automation,* Raleigh, 1987.

[81] Shimon Ullman and Ronen Basri, "The Alignment Method with Smooth Surfaces." *MIT AI Memo* 1060, July 1988.

[82] Shimon Ullman and Amnon Sha'ashua, "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network." *MIT AI Memo* 1061, July 1988.

[83] Shimon Ullman, "Aligning Pictorial Descriptions: An approach to object recognition." *Cognition,* vol 32, No. 3, August, 1989.

[84] D. L. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows" *AI-TR*-271, MIT Artificial Intelligence Lab, MIT, 1972.

[85] D. L. Waltz, "Shedding Light on shadows" *MIT AI-TR*-281, P. H. Winston (ed), May 1973.

[86] D. L. Waltz, "Understanding line drawings of scenes with shadows" *The Psychology of Computer Vision,* P. H. Winston (ed), McGraw-Hill, New York, pp. 19-91, 1975.

[87] P. H. Winston, "Machine Vision." *The Psychology of Computer Vision,* P. H. Winston (ed), McGraw-Hill, New York, 1975.

[88] K. M. Yip, "KAM: Automatic Planning and Interpretation of Numerical Experiments Using Geometrical Methods." *MIT AI-TR*-1163, August 1989.

[89] Steven W. Zucker, "The Emerging Paradigm of Computational Vision." *Ann. Rev. Comput. Sci.* 2:69-89, 1987.